

---

# Regularizing Hidden States Enables Learning Generalizable Reward Model for LLMs

---

Rui Yang<sup>1</sup> Ruomeng Ding<sup>2</sup> Yong Lin<sup>3 4</sup> Huan Zhang<sup>1</sup> Tong Zhang<sup>1</sup>

<sup>1</sup>University of Illinois Urbana-Champaign, <sup>2</sup>Georgia Institute of Technology,

<sup>3</sup>Princeton University, <sup>4</sup>Princeton Language and Intelligence

yangrui.thu2015@gmail.com, rmding@gatech.edu, yl7690@princeton.edu

huan@huan-zhang.com, tongzhang@tongzhang-ml.org

## Abstract

Reward models trained on human preference data have been proven to effectively align Large Language Models (LLMs) with human intent within the framework of reinforcement learning from human feedback (RLHF). However, current reward models have limited generalization capabilities to unseen prompts and responses, which can lead to an unexpected phenomenon known as reward over-optimization, resulting in a decline in actual performance due to excessive optimization of rewards. While previous research has advocated for constraining policy optimization, our study introduces a novel approach to enhance the reward model’s generalization ability against distribution shifts by regularizing the hidden states. Specifically, we retain the base model’s language model head and incorporate a suite of text-generation losses to preserve the hidden states’ text-generation capabilities, while concurrently learning a reward head behind the same hidden states. Our experimental results demonstrate that the introduced regularization technique markedly improves the accuracy of learned reward models across a variety of out-of-distribution (OOD) tasks and effectively alleviates the over-optimization issue in RLHF, offering a more reliable and robust preference learning paradigm [\[1\]](#).

## 1 Introduction

Pretrained large models have showcased impressive capabilities across diverse fields [\[1, 2, 3, 4, 5\]](#). A notable trend in recent research is ensuring that large models align with human values and mitigate potentially harmful behaviors [\[6, 7, 8, 9, 10\]](#). Alignment methods are crucial in achieving this objective, with two primary approaches being supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF) [\[7, 8\]](#). SFT directly finetunes the model using prompt and response pairs, proving to be a straightforward and efficient alignment technique [\[11, 12, 13\]](#). Differently, RLHF begins by learning a reward model from user preferences and then employs reinforcement learning to optimize the language model to maximize rewards. A significant advantage of RLHF is its potential to generalize the reward model to unseen prompt-response pairs, effectively leveraging large volumes of unlabeled data [\[8, 14\]](#).

Despite the empirical success of RLHF, the challenge of training a reliable and generalizable reward model for unseen data remains an open problem. A well-known failure mode of reward model is known as "overoptimization" or "reward hacking" [\[15, 16, 17, 18\]](#), where policy optimization seemingly improves the proxy reward model but actually degrades the true reward function. [\[17\]](#) demonstrated in a synthetic setup that increasing the size of the reward model and the volume of training data can mitigate this overoptimization issue. However, such scaling is not always feasible

---

<sup>1</sup>Code and open-source reward models are available at <https://github.com/YangRui2015/Generalizable-Reward-Model>

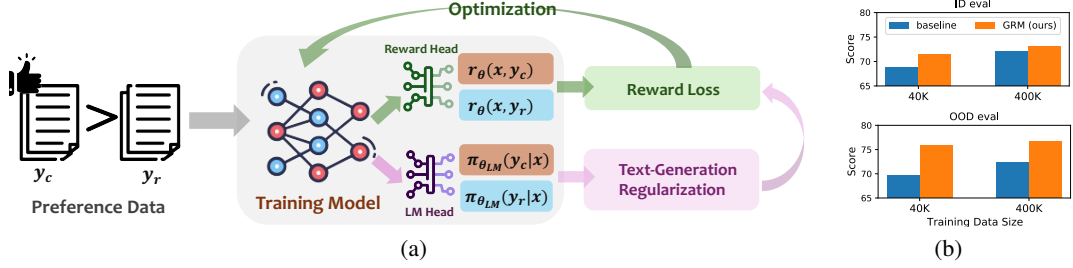


Figure 1: (1) Illustration of GRM. Given preference data pairs  $(x, y_c, y_r)$ , the reward head  $r_\theta$  minimizes the reward loss in Eq 1 while the language model (LM) head  $\pi_{\theta_{LM}}$  minimizes a suite of text-generation losses introduced in Sec 3.2 (2) Performance of GRM and the vanilla reward model on in-distribution (ID) task (Unified-Feedback) and average results of OOD tasks (HHH-Alignment and MT-Bench). Compared with the baseline reward model, GRM generalizes better on OOD tasks, with a larger advantage when the dataset size is relatively small.

in many realistic scenarios. To address this, a series of studies have been conducted, focusing either on enhancing the reward model with ensemble techniques [18, 19, 20, 21] or on constrained policy optimization [22, 23, 24, 25]. The latter paradigm is related to the offline RL literature [26, 27, 28, 29, 30, 31], which involves limiting the policy distribution to be close to the training data distribution. Among these, improving the generalization ability of reward models presents a fundamental and promising direction that can be studied independently from enhancements in policy optimization. Nevertheless, previous methods [18, 32] requiring training multiple reward models may be resource-intensive for the practical application of large models.

In this study, we present a lightweight yet effective solution designed to enhance the reward model’s generalization ability against distribution shifts. Previous research [33] has theoretically shown that a randomly initialized head can distort pre-trained features, thereby negatively impacting out-of-distribution (OOD) performance. Inspired by this finding, we propose to regularize the feature during fine-tuning for preference learning using an adversarial regularizer, which derives a suite of text-generation losses. To this end, we introduce **Generalizable Reward Model (GRM)**, which retains the base model’s language model head and regularizes the hidden states of the reward model by incorporating text-generation losses. This approach makes better use of the preference learning data while preserving the text generation capabilities of the hidden states. Notably, GRM does not necessitate training multiple reward models or relying on additional training data.

In our experiments, GRM substantially improves the evaluation accuracy of the reward model OOD evaluation datasets, demonstrating its superior ability to generalize learned preferences to unseen prompt and response pairs. Moreover, GRM consistently improves the performance of both 2B and 7B reward models, with a more pronounced improvement observed when the data size is limited. We also demonstrate that GRM can markedly enhance the performance of best-of- $n$  (BoN) sampling and PPO [34], effectively mitigating the overoptimization problem. These results highlight the potential of the GRM to serve as a more reliable proxy reward model for human preferences.

To conclude, our primary contributions are as follows:

- We propose GRM, a novel approach that employs text-generation regularization on the hidden states to enhance the generalization ability of reward models.
- Our study validates the effectiveness of all three types of text-generation regularization for GRM, identifying the SFT regularization as the most effective and stable solution.
- Our empirical results show that GRM significantly improves the accuracy of reward models across various OOD tasks. Furthermore, it consistently enhances the performance of RLHF, effectively alleviating the overoptimization problem.

## 2 Background

Typically, reinforcement Learning from Human Feedback (RLHF) involves reward modeling and policy optimization, with Best-of- $n$  Sampling (BoN) and Proximal Policy Optimization (PPO) being two commonly used methods for policy optimization.

**Reward Modeling.** Generally, reward modeling is based on the Bradley-Terry model [35], which aims to distinguish between the chosen response  $y_c$  and the rejected response  $y_r$  given the prompt  $x$ :

$$\mathcal{L}_{\text{reward}}(\theta) = -\mathbb{E}_{(x, y_c, y_r) \sim D} [\log(\sigma(r_\theta(x, y_c) - r_\theta(x, y_r)))], \quad (1)$$

where  $r_\theta(x, y)$  represents the reward score for prompt  $x$  and output  $y$  with model parameters  $\theta$ .  $\sigma(\cdot)$  is the sigmoid function. By minimizing this loss function, the reward model assigns higher scores to outputs preferred by humans. Subsequently, the trained reward model can be used to guide the optimization of the language model.

**Best-of- $n$  Sampling (BoN).** BoN generates  $n$  samples from the policy model, denoted as  $Y_{\text{gen}}$ , and then selects the best one based on scores provided by a reward model. BoN can be used for inference-time improvement or iterative optimization [36, 37, 38].

$$y_{\text{BON}}(x) = \arg \max_{y \in Y_{\text{gen}}} r_\theta(x, y). \quad (2)$$

**Proximal Policy Optimization (PPO).** PPO is a widely adopted method for RLHF in optimizing language models [16, 8, 39]. PPO learns a policy by maximizing a reward objective with a KL divergence penalty with coefficient  $\eta$ :

$$r_{\text{total}} = r_\theta(x, y) - \eta \text{KL}(\pi_{\text{PPO}}(y|x) \parallel \pi_{\text{SFT}}(y|x)), \quad (3)$$

where the KL penalty ensures that the optimized policy does not deviate significantly from the SFT policy to maintain the reliability of the reward model.

**Overoptimization.** Although the learned proxy reward model aims to approximate human preference, it may not consistently reflect authentic human preferences, potentially resulting in *over-optimization* [17, 18]. This issue emerges when the proxy reward model becomes overly optimized, causing the policy model to overfit certain erroneous patterns. Ultimately, this issue can diminish the model’s alignment with actual human preferences, highlighting the need to ensure the reward model’s robustness and reliability.

### 3 Method

In the common practice of training a reward model [8, 39, 40], reward models are initialized using a pretrained or SFT finetuned backbone, along with a randomly initialized reward head to predict the scores for prompt-response pairs. It’s important to note that the backbone and original language model head are trained on a diverse range of datasets for text generation, which is distinct from the preference learning tasks. Under the task shift, the randomly initialized reward head can distort the pretrained features, thereby reducing the OOD generalization performance, as observed by [33]. We also confirm this impact on preference learning in Appendix C.1

To improve the reward model’s generalization capability against distribution shifts, we propose a lightweight yet effective solution, Generalizable Reward Model (GRM). This model employs a suite of text-generation regularizations for the hidden states. More specifically, GRM employs a structure as illustrated in Fig 1, with one language model (LM) head and one reward head sharing the same hidden states. The reward head is trained to minimize the reward loss  $\mathcal{L}_{\text{reward}}$  in Eq 1, while the LM head is trained to maintain the text-generation ability of the hidden states during preference learning. Consequently, we define the overall loss function as follows:

$$\mathcal{L}_{\text{total}} = (1 - \alpha)\mathcal{L}_{\text{reward}} + \alpha\mathcal{L}_{\text{reg}}. \quad (4)$$

Here,  $\alpha$  is the coefficient that balances the reward loss and the regularization. We will derive potential forms of the regularization term below.

#### 3.1 Theoretical Motivation

To derive the potential formulation of the regularization term, we consider the following adversarial optimization problem: learning a reward model against an adversarial policy.

$$\theta = \arg \min_{\theta} \left\{ \mathcal{L}_{\text{reward}}(\theta) + \gamma \max_{\pi} J(\theta, \pi) \right\}, \quad (5)$$

where  $\gamma > 0$  is a coefficient. This objective is also considered by recent studies [24, 25] aiming to enhance DPO. Differently, we adopt it to learn a generalizable reward model.

The insight of Eq 5 is that we can enhance the robustness of the reward model by considering an adversarial policy  $\pi$  from a certain policy class. The term for policy optimization  $J(\theta, \pi)$  can have various formulations, but a KL divergence-regularized objective is generally used in training the policy [16, 8]. Moreover, it has an advantageous property that the inner optimization problem has an analytical solution, which can simplify the problem.

$$J(\theta, \pi) = \mathbb{E}_{x \sim D, y \sim \pi(\cdot|x)} [r_\theta(x, y)] - \beta \mathbb{E}_{x \sim D} [\text{KL}(\pi(\cdot|x) \parallel \pi_{\text{ref}}(\cdot|x))], \quad (6)$$

where  $\beta > 0$  is a regularization coefficient and  $\pi_{\text{ref}}$  is the reference model. We denote the analytical solution of  $J(\theta, \pi)$  as  $\pi_\theta^*$ . Incorporating  $\pi_\theta^*$  into Eq 5, we can transform the min-max optimization problem into a standard optimization problem under certain assumptions:

$$\theta = \arg \min_{\theta} \{(1 - \alpha) \mathcal{L}_{\text{reward}}(\theta) + \alpha_{\text{DPO}} \mathcal{L}_{\text{DPO}}(\pi_\theta^*) + \alpha_{\text{SFT}} \mathcal{L}_{\text{SFT}}(\pi_\theta^*)\} \quad (7)$$

Detailed derivation is deferred to Appendix A. Here,  $\mathcal{L}_{\text{DPO}}$  is the same as the DPO objective [41] and  $\mathcal{L}_{\text{SFT}}$  is the SFT objective that maximizes the probability of chosen responses. Notably, the two regularization terms originate from different sources:  $\mathcal{L}_{\text{DPO}}$  stems from the reward loss, while  $\mathcal{L}_{\text{SFT}}$  is derived from the adversarial term. This may explain why SFT regularization proves more beneficial than DPO regularization in our empirical results. Motivated by Eq 7, we relax the relationship between  $r_\theta$  and  $\pi_\theta^*$  and propose learning a reward model parameterized by  $\theta$  and a language model head parameterized by  $\theta_{\text{LM}}$ , both sharing the same hidden states. A discussion of this design can be found in Appendix A. Below, we detail three practical implementations.

### 3.2 Text-Generation Regularization

Inspired by Eq 7, we train the LM head to minimize text-generation losses, such as DPO and SFT losses, as the regularization term for GRM. To independently study the effectiveness of these two regularizations and reduce GPU memory usage, we introduce three practical implementations: DPO regularization, DPO without reference regularization, and SFT regularization.

**DPO Regularization.** By setting  $\alpha_{\text{DPO}} = \alpha$  and  $\alpha_{\text{SFT}} = 0$  in Eq 7, we can directly adopt the DPO loss as a regularization term for GRM to regularize the hidden states:

$$\mathcal{L}_{\text{DPO}}(\theta_{\text{LM}}) = -\mathbb{E}_{(x, y_c, y_r) \sim D} \left[ \log \sigma \left( \beta \log \left( \frac{\pi_{\theta_{\text{LM}}}(y_c | x)}{\pi_{\text{ref}}(y_c | x)} \right) - \beta \log \left( \frac{\pi_{\theta_{\text{LM}}}(y_r | x)}{\pi_{\text{ref}}(y_r | x)} \right) \right) \right], \quad (8)$$

where  $\pi_{\text{ref}}$  is the base model serving as the reference model, and  $\pi_{\theta_{\text{LM}}}$  is our optimized policy.  $\beta$  is a coefficient that controls the KL penalty between  $\pi_{\theta_{\text{LM}}}$  and  $\pi_{\text{ref}}$ . Notably,  $\pi_{\theta_{\text{LM}}}$  shares the same base model with the reward model  $r_\theta$ , except for the output layer.

**DPO Regularization w/o Reference Model.** While straightforward, the use of a reference model in DPO regularization can be memory-intensive for large models. To address this, and inspired by prior works that eliminate the need for reference model [42, 43], we introduce the DPO regularization without a reference model, denoted as  $\mathcal{L}_{\text{DPO-no ref}}$ . This method reduces the need for large GPU memory during training. The loss function  $\mathcal{L}_{\text{DPO-no ref}}$  is defined as:

$$\mathcal{L}_{\text{DPO-no ref}}(\theta_{\text{LM}}) = -\mathbb{E}_{(x, y_c, y_r) \sim D} \left[ \log \sigma \left( \beta \log \left( \frac{\pi_{\theta_{\text{LM}}}(y_c | x)}{\pi_{\theta_{\text{LM}}}(y_r | x)} \right) \right) \right]. \quad (9)$$

**SFT Regularization.** By setting  $\alpha_{\text{DPO}} = 0$  and  $\alpha_{\text{SFT}} = \alpha$  in Eq 7, we can simplify the regularization term to SFT regularization, thereby reducing the computational cost. This method only maximizes the probability of the chosen responses:

$$\mathcal{L}_{\text{SFT}}(\theta_{\text{LM}}) = -\mathbb{E}_{(x, y_c) \sim D} [\log \sigma(\beta \log(\pi_{\theta_{\text{LM}}}(y_c | x)))]. \quad (10)$$

This equation differs slightly from the standard SFT objective to maintain coherence with the above two cases within the regularization suite and avoid the need for hyperparameter adjustments for  $\alpha$ . Please refer to Appendix C.3 for a discussion.

### 3.3 Advantages of GRM

In summary, GRM offers three key advantages: (1) **Mitigating feature distortion.** The application of text-generation loss helps maintain the text-generation ability of the base model and prevents

excessive feature distortion. Simultaneously, it also adapts the model to the data distribution of preference learning. **(2) Prevention of Overfitting.** The text-generation regularization derived from an adversarial training objective helps prevent the reward model from overfitting to certain spurious features, which can be detrimental to OOD generalization. This effect becomes more pronounced when the preference data includes erroneous comparison pairs or when the dataset size is limited. **(3) Efficiency.** GRM is an efficient solution that does not require training multiple reward models or additional training data. Additionally, different choices of loss type entail varying memory and computational costs. Interestingly, we find that the simplest option, SFT regularization, proves to be the most stable choice.

## 4 Experimental Setup

**Datasets.** For training reward models, we leverage the Unified-Feedback dataset<sup>2</sup>, which stands as one of the largest collections of pairwise feedback datasets. In Section 5.1, we train all reward models on a subset of 400K and 40K samples from the Unified-Feedback dataset and evaluate them on the hold-out 8K eval set. In addition, for evaluating model performance on out-of-distribution (OOD) preference data, we utilize datasets such as HHH-Alignment<sup>3</sup> [44], MT-Bench Human Judgements<sup>4</sup> [45], and RewardBench [46]. The HHH-Alignment dataset evaluates language models on helpfulness, honesty, and harmlessness, while the MT-Bench dataset contains human preferences for model responses to MT-bench questions. Besides, RewardBench is a new benchmark designed to evaluate the capabilities and safety of reward models. We consider HHH-Alignment, MT-Bench, and RewardBench as OOD evaluation tasks because the prompt and response distributions differ from our training distribution. For the RLHF experiments in Section 5.2 we downsample 20K data from Unified-Feedback for training reward models and optimizing the PPO policy, and another 1K data for evaluating BoN or the learned PPO policy.

**Base Models.** In the preference learning experiments, our base models include gemma-2B-it [47] and Mistral-7B-Instruct-v0.2 [48]. For the RLHF experiments, gemma-2B-it serves as the policy model for both BoN and PPO experiments, whereas the gold reward model<sup>5</sup> is a 7B human preference model finetuned using the entire Unified-Feedback dataset.

**Baselines.** We compare the performance of GRM with several baselines, including *Baseline Classifier* trained using the original reward loss in Eq 1; *Frozen Classifier* that fixes the base model’s feature and only finetunes a nonlinear classification head; *Margin* that adds an additional margin in the original reward loss [10, 39]; *Label Smooth* that mitigate the overfitting problem by penalizing overconfident model outputs [39]; *Ensemble* method with a group of 3 reward models [18] to calculate the average or minimum values as rewards. In addition, for RewardBench, we present the performance of several existing open-source state-of-the-art reward models for better reference, including PairRM [49], Starling-RM-7B/34B [50], and UltraRM-13B [51]. For more experimental details and additional results, please refer to Appendix B and Appendix C respectively.

## 5 Evaluation Results

We present a comprehensive evaluation of GRM, utilizing both in-distribution (ID) and out-of-distribution (OOD) datasets, as well as existing benchmarks for reward models. Furthermore, we explore the impact of GRM on the overoptimization issue in RLHF. Our primary findings can be summarized as follows:

- GRM significantly enhances the generalization capability of reward models, resulting in substantial improvements on both ID and various OOD evaluation sets (Section 5.1).
- All three types of text-generation regularization losses can improve the generalization performance, with the SFT regularization being the most effective and stable (Section 5.1).
- GRM demonstrates robustness in the limited dataset setting, outperforming baselines by an even larger margin (Section 5.1).

<sup>2</sup><https://huggingface.co/datasets/llm-blender/Unified-Feedback>

<sup>3</sup>[https://huggingface.co/datasets/HuggingFaceH4/hhh\\_alignment](https://huggingface.co/datasets/HuggingFaceH4/hhh_alignment)

<sup>4</sup>[https://huggingface.co/datasets/lmsys/mt\\_bench\\_human\\_judgments](https://huggingface.co/datasets/lmsys/mt_bench_human_judgments)

<sup>5</sup>[reward-model-Mistral-7B-instruct-Unified-Feedback](#)



- GRM effectively mitigates the overoptimization issue in both BoN and PPO (Section 5.2).
- GRM also exhibits robustness against label noise in the preference dataset (Section 5.2).

Table 1: Results on ID and OOD evaluation with **400K training data** from Unified-Feedback. The best performance in each task is in bold and the second best one is underlined.

| Reward Model                   | Unified Feedback | HHH Alignment | MT Bench    |
|--------------------------------|------------------|---------------|-------------|
| Classifier (Frozen)            | 63.8             | 66.4          | 69.5        |
| Classifier (baseline)          | 72.1             | 73.4          | 71.2        |
| Classifier + margin            | 72.0             | 75.0          | 72.6        |
| Classifier + label smooth      | 71.5             | 72.1          | 71.2        |
| Classifier + Ensemble          | 72.8             | 76.8          | <b>73.7</b> |
| <b>GRM w/ dpo (ours)</b>       | <u>73.8</u>      | 79.2          | <u>73.4</u> |
| <b>GRM w/ dpo-noref (ours)</b> | <b>73.9</b>      | <u>79.7</u>   | 73.0        |
| <b>GRM w/ sft (ours)</b>       | 73.2             | <b>79.8</b>   | <u>73.4</u> |

Table 2: Results on ID and OOD evaluation with **40K training data** from Unified-Feedback. The best performance in each task is in bold and the second best one is underlined.

| Reward Model                   | Unified Feedback | HHH Alignment | MT Bench    |
|--------------------------------|------------------|---------------|-------------|
| Classifier (Frozen)            | 63.9             | 68.6          | 68.2        |
| Classifier (baseline)          | 68.8             | 70.3          | 69.1        |
| Classifier + margin            | 69.6             | 69.8          | 71.0        |
| Classifier + label smooth      | 68.5             | 68.8          | 71.9        |
| Classifier + Ensemble          | 69.9             | 72.2          | 71.1        |
| <b>GRM w/ dpo (ours)</b>       | 70.2             | 71.6          | 71.3        |
| <b>GRM w/ dpo-noref (ours)</b> | <u>71.4</u>      | <u>76.6</u>   | <u>72.1</u> |
| <b>GRM w/ sft (ours)</b>       | <b>71.5</b>      | <b>78.7</b>   | <b>73.0</b> |

## 5.1 Evaluation on Reward Modeling

**ID and OOD Evaluation.** The results, shown in Table 1 and Table 2, illustrate the evaluation performance of different reward modeling methods using the gemma-2B-it base model on both ID (Unified-Feedback) and OOD (HHH-Alignment and MT-Bench) datasets. Regardless of the size of the training data (400K or 40K), our proposed method, GRM, with three types of regularizations, consistently outperforms the baseline models on both the ID evaluation set and the two OOD datasets. For instance, GRM w/ sft with 400K training data enhances the baseline from 72.1 to 73.2 in ID score, and improves the HHH-Alignment score from 73.4 to 79.8 and the MT-Bench score from 71.2 to 73.4. Notably, the improvement in OOD performance is significantly larger than that in ID. These results suggest that the GRM methods are highly effective in evaluating unseen preference data, demonstrating substantially robust generalization capabilities.

Regarding other baseline models, the Frozen classifier, which maintains its base model’s parameters, exhibits the lowest ID and OOD scores. This suggests that the pretrained features of the base model alone are insufficient for effective preference learning, emphasizing the importance of fine-tuning the base model’s features to the preference task. Furthermore, the margin loss and label smoothing techniques do not consistently improve the ID and OOD tasks, whereas the ensemble baseline consistently enhances both ID and OOD scores. Despite requiring the training of multiple reward models, ensemble-based methods still do not surpass GRM, particularly when learning from a 40K training set. These results highlight the substantial improvement and generalization capability of GRM in preference learning.

**Comparison of Different Regularizations.** As observed in Table 1, when the training dataset is sufficiently large, GRM with three types of regularizations (namely GRM w/ dpo, GRM w/ dpo-noref, and GRM w/ sft) perform comparably. This demonstrates that GRM is robust to the choice of regularization type when the dataset is large. However, in Table 2, where the training data is limited to 40K, a clear trend emerges: GRM w/ sft outperforms GRM w/ dpo-noref, which in turn outperforms GRM w/ dpo, on both the ID and OOD scores. Interestingly, the simplest form of regularization, SFT regularization, not only requires the lowest training cost but also yields the most stable overall results. Consequently, we adopt it as the default choice for our subsequent study.

**Results on RewardBench.** In Table 3 and Table 4, we evaluate GRM and various baselines on RewardBench across chat, chat-hard, safety, and reasoning task groups. We consider a variant of GRM with a linear reward head instead of the default nonlinear reward head as detailed in Appendix B. In Table 3, the 7B baseline matches the score of Starling-RM-7B [50], while GRM (linear) w/ sft demonstrates a considerable improvement, increasing the average score from 76.3 to 79.5. Comparing variants of GRM, we can conclude that: (1) SFT regularization performs better than the DPO w/o reference model regularization, and (2) GRM with a linear head achieves a better overall score than that with a nonlinear head, especially in the challenging reasoning task group.

Table 3: Results on RewardBench with **400K training data** from Unified-Feedback.

| Reward model                     | Average     | Chat        | Chat-Hard   | Safety      | Reasoning   |
|----------------------------------|-------------|-------------|-------------|-------------|-------------|
| PairRM                           | 58.7        | 90.2        | 53.0        | 31.5        | 60.0        |
| Starling-RM-7B                   | <u>76.2</u> | 98.0        | 43.4        | 88.6        | 74.6        |
| Starling-RM-34B                  | <b>84.0</b> | 96.9        | 59.0        | 89.9        | 90.3        |
| UltraRM-13B                      | 69.8        | 96.1        | 55.3        | 45.8        | 82.0        |
| Base Model: Gemma 2b it          |             |             |             |             |             |
| Classifier (baseline)            | 68.2        | 95.5        | 38.0        | 73.8        | 65.3        |
| Classifier + margin              | 70.2        | 95.8        | 38.4        | 73.9        | <u>72.5</u> |
| Classifier + label smooth        | 70.6        | 94.4        | 37.3        | 73.2        | <b>77.4</b> |
| Classifier + Ensemble            | <u>71.0</u> | <b>98.0</b> | 37.5        | 77.3        | 71.3        |
| GRM (linear) w/ dpo noref (ours) | 70.2        | 96.7        | 39.0        | 76.4        | 68.5        |
| GRM (linear) w/ sft (ours)       | <b>71.5</b> | 96.1        | <u>40.1</u> | <b>80.3</b> | 69.3        |
| GRM w/ dpo noref (ours)          | 70.2        | 95.8        | <u>40.1</u> | <u>78.7</u> | 66.2        |
| GRM w/ sft (ours)                | 70.8        | <u>97.8</u> | <b>42.1</b> | 77.9        | 65.2        |
| Base Model: Mistral 7b Instruct  |             |             |             |             |             |
| Classifier (baseline)            | 76.3        | 96.6        | 52.4        | 86.7        | 69.5        |
| Classifier + margin              | 74.5        | 96.4        | 51.5        | 85.3        | 64.8        |
| Classifier + label smooth        | 76.3        | 97.2        | 49.8        | 85.8        | 72.3        |
| Classifier + Ensemble            | 76.6        | 96.6        | 51.8        | 85.1        | 73.0        |
| GRM (linear) w/ dpo noref (ours) | <u>78.3</u> | <b>98.0</b> | 53.3        | <b>86.4</b> | <u>75.3</u> |
| GRM (linear) w/ sft (ours)       | <b>79.5</b> | <u>97.8</u> | <u>54.6</u> | <u>86.3</u> | <b>79.2</b> |
| GRM w/ dpo noref (ours)          | 78.0        | <u>97.8</u> | 54.0        | 85.7        | 74.4        |
| GRM w/ sft (ours)                | 77.6        | <b>98.0</b> | <b>55.3</b> | 85.8        | 71.2        |

Table 4: Results on RewardBench with **40K training data** from Unified-Feedback.

| Reward model                     | Average     | Chat        | Chat-Hard   | Safety      | Reasoning   |
|----------------------------------|-------------|-------------|-------------|-------------|-------------|
| Base Model: Gemma 2B it          |             |             |             |             |             |
| Classifier (baseline)            | 64.5        | <u>95.8</u> | 37.3        | 59.9        | 64.8        |
| Classifier + margin              | 66.1        | <b>97.2</b> | 37.5        | 56.8        | 72.7        |
| Classifier + label smooth        | 61.1        | 91.6        | 39.0        | 53.8        | 60.2        |
| Classifier + Ensemble            | 65.2        | 96.1        | 38.2        | 58.8        | <u>67.6</u> |
| GRM (linear) w/ dpo noref (ours) | 61.7        | 94.7        | 38.4        | 62.5        | 51.2        |
| GRM (linear) w/ sft (ours)       | <b>69.5</b> | 94.7        | <u>40.8</u> | 65.4        | <b>77.0</b> |
| GRM w/ dpo noref (ours)          | 66.6        | 92.5        | 39.9        | <b>72.5</b> | 61.4        |
| GRM w/ sft (ours)                | <u>66.8</u> | 94.1        | <b>41.9</b> | <u>69.5</u> | 61.5        |
| Base Model: Mistral 7B Instruct  |             |             |             |             |             |
| Classifier (baseline)            | 68.2        | 89.7        | 50.7        | 74.7        | 57.9        |
| Classifier + margin              | 62.8        | 89.7        | 47.1        | 70.7        | 43.6        |
| Classifier + label smooth        | 72.1        | 94.1        | 47.1        | 67.5        | 79.7        |
| Classifier + Ensemble            | 69.3        | 89.6        | 50.2        | 72.7        | 59.0        |
| GRM (linear) w/ dpo noref (ours) | 77.8        | 96.9        | 52.9        | <u>82.7</u> | 78.8        |
| GRM (linear) w/ sft (ours)       | 78.3        | 96.7        | 52.4        | 81.5        | <b>82.5</b> |
| GRM w/ dpo noref (ours)          | <b>78.6</b> | <b>97.8</b> | <b>54.6</b> | 82.0        | <u>79.9</u> |
| GRM w/ sft (ours)                | <u>78.4</u> | <u>97.2</u> | <u>54.2</u> | <b>83.6</b> | 78.6        |

Regarding the baselines, consistent with previous results, the margin loss and label smoothing do not provide a coherent improvement over the baseline. While ensemble methods effectively improve upon the baseline, they still underperform GRM. Overall, these results demonstrate that GRM is a strong contender in reward modeling tasks, exhibiting superior performance across various benchmarks.

**Comparison of Different Dataset Sizes.** Another noteworthy observation is that GRM exhibits greater robustness to the size of the training dataset compared to baselines. For instance, in Table 1 and Table 2, when the training data size decreases from 400K to 40K, the baseline’s HHH Alignment score and MT-Bench score drop from 73.4 and 71.2 to 70.3 and 69.1, respectively. In contrast, GRM with SFT regularization only slightly drops from 79.8 and 73.4 to 78.7 and 73.0, respectively. This

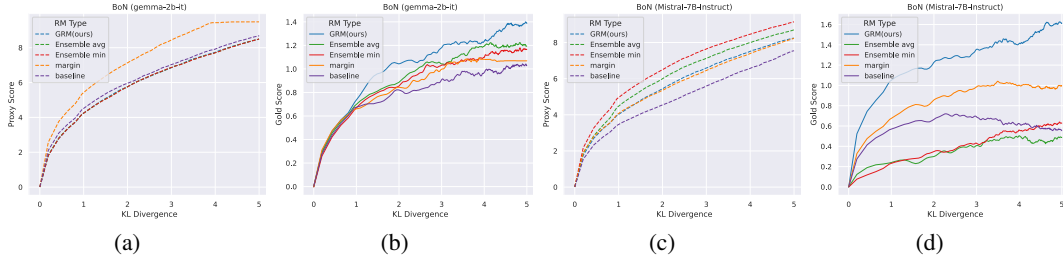


Figure 2: Proxy scores and gold scores of BoN experiments for base models of (a)(b) gemma-2b-it and (c)(d) Mistral-7B-Instruct. Proxy and gold scores are in dashed and solid curves, respectively. Rewards are normalized to start from 0. GRM demonstrates a robust ability to select the best response aligned with the gold rewards as the KL Divergence increases.

trend is consistent in Table 3 and Table 4. Specifically, for the Mistral 7B Instruct base model, the baseline’s average score drops from 76.3 to 68.2 when learning from 40K training data, while GRM (linear) w/ sft only drops from 79.5 to 78.3. These findings suggest that the prior reward training paradigms are sensitive to smaller dataset sizes. In contrast, GRM is robust even with a limited dataset.

**Full Parameter Training Results on a Larger Dataset.** To further demonstrate the effectiveness of GRM, we trained the GRM using the llama3-8b-instruct model [52]. We perform a full parameter fine-tuning for 1 epoch on one of the largest open-source preference datasets [6]. Our results, presented in Table 5, highlight the considerable potential of scaling GRM to larger models and datasets. Especially, GRM outperforms a 34B reward model and even GPT-4 as a judge. It is worth noting that the GRM significantly improves the performance of the 8B reward model from 84.7 to 87.0, using the same base model and training data as FsfairX-LLaMA3-RM-v0.1 [40]. This improvement is particularly remarkable in the challenging ‘Reasoning’ section.

Table 5: Results of full parameter training on RewardBench.

| Reward model           | Average     | Chat        | Chat-Hard   | Safety      | Reasoning   |
|------------------------|-------------|-------------|-------------|-------------|-------------|
| GRM (Ours, 8B)         | <b>87.0</b> | 98.6        | 67.8        | <b>89.4</b> | <b>92.3</b> |
| gpt-4-0125-preview     | 85.9        | 95.3        | 74.3        | 87.2        | 86.9        |
| gpt-4-turbo-2024-04-09 | 85.1        | 95.3        | <b>75.4</b> | 87.1        | 82.7        |
| FsfairX-LLaMA3-RM-8B   | 84.7        | <b>99.4</b> | 65.1        | 87.8        | 86.4        |
| Starling-RM-34B        | 82.7        | 96.9        | 57.2        | 88.2        | 88.5        |

## 5.2 Evaluation on RLHF

**Best-of- $n$  Sampling (BoN).** Fig 2 presents the results of BoN sampling for base models of sizes 2B and 7B. For all BoN experiments, we utilize a 20K subset from the Unified-Feedback dataset, labeled by the gold reward model, to train proxy reward models. Following the [17, 18], we conduct BoN sampling on a 1K held-out test set from  $n$  responses for each prompt, based on the scores of the trained proxy model. The selected responses are then scored using the gold reward model, and their gold scores are averaged over the 1K test prompts. The average gold score reveals the true quality of the responses selected by the proxy reward models. We set the KL Divergence from 0 to 5, corresponding to the number of responses  $n$  ranging from 1 to 405 for each prompt, according to the equation  $KL_{BoN} = \log n - \frac{n-1}{n}$ . Ideally, a good proxy reward model should yield larger average proxy and gold scores as the KL increases. However, the average gold scores of some baseline methods plateau or even drop after  $KL > 4$ , such as the baseline reward model in Fig 2(d), despite their proxy scores continuing to increase in Fig 2(c). This suggests that these reward models suffer from the overoptimization issue.

<sup>6</sup>[https://huggingface.co/datasets/hendrydong/preference\\_700K](https://huggingface.co/datasets/hendrydong/preference_700K)



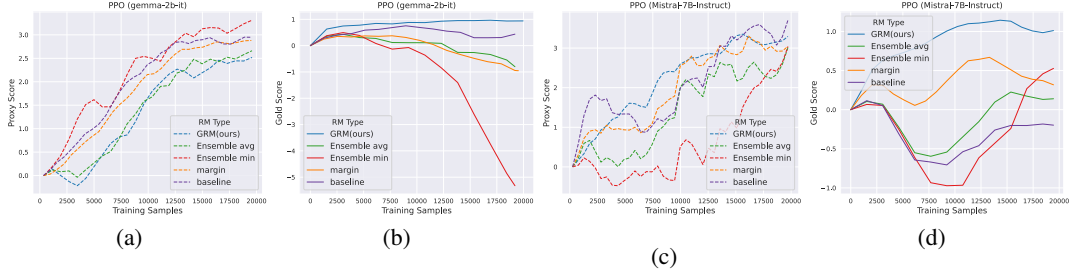


Figure 3: Proxy scores and gold scores of PPO experiments for reward model based on (a)(b) gemma-2b-it and (c)(d) Mistral-7B-Instruct. All rewards are normalized to start from 0.

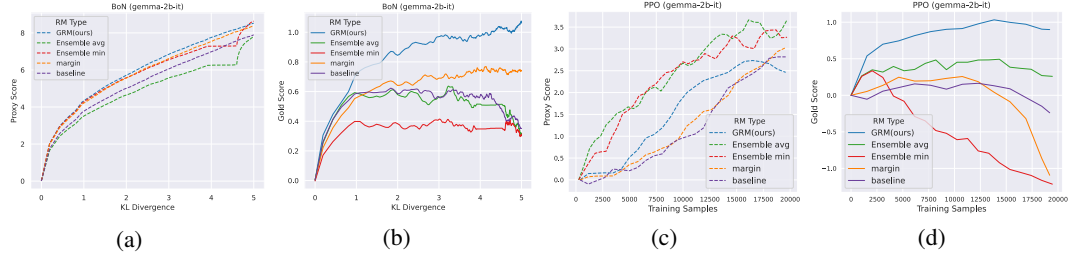


Figure 4: Proxy scores and gold scores of (a)(b) BoN experiments and (c)(d) PPO experiments with 25% label noise. All rewards are normalized to start from 0.

In contrast, GRM consistently demonstrates an increase in both the proxy score and gold score, indicating that it effectively mitigates over-optimization. This advantage is more pronounced in the 7B base model, where GRM achieves an average gold score of 1.5, while the baseline reward model only reaches a score of 0.5. Regarding other baselines, we find that the margin loss and ensemble methods (especially the 'min' strategy) contribute to the robustness of the reward model. However, they still do not compare favorably with GRM. These results underscore the strong potential of GRM to serve as a reliable and robust proxy reward model for RLHF.

**Proximal Policy Optimization (PPO).** To investigate whether GRM can effectively mitigate the overoptimization issue in PPO, we further employ those 2B and 7B reward models obtained from the BoN experiments to fine-tune the policy model (gemma-2b-it) using PPO. The training and evaluation datasets are identical to the BoN experiments. We train PPO for one epoch on the training set, comprising 20K training samples. As depicted in Fig 3, PPO exhibits a stronger tendency to hack the learned reward models compared to BoN. The gold scores of baseline methods begin to decline early in the training process, while their proxy scores increase, indicating a clear overoptimization issue. In contrast, GRM demonstrates superior robustness in terms of the gold score, which rises with the increase in proxy scores. This validates that GRM can effectively alleviate overoptimization for PPO. Please refer to Appendix D for a clear comparison of the results generated by PPO.

**Robustness to Label Noise.** Human preference data typically contains around 20 to 30% noise, as highlighted in previous studies [39]. Such inconsistent preference data can render the reward model less generalizable [32, 53] and hinder policy learning [54, 55, 56], leading to a performance decline. To evaluate the robustness of GRM against label noise, we incorporate a 25% label noise into the 20K training data for all proxy reward models. The results are depicted in Fig 4. Most gold scores expose a more severe over-optimization issue, as compared to the results in Fig 2(b) and Fig 3(b), indicating that those reward models are heavily overfitting under the noisy label setting. On the contrary, GRM exhibits superior robustness under noisy conditions, consistently achieving a peak gold score over 1.0 without a significant decline. This demonstrates that GRM is highly accurate and robust at measuring the sample quality, even in the presence of noise within the training data.

## 6 Related Works

**Reward Modeling.** Reward models, trained on human preference data, are crucial in guiding RLHF [8, 57] or prompt optimization [58]. Recent studies have concentrated on developing advanced reward models to improve the performance of LLMs in RLHF. One approach involves enhancing reward modeling by improving the quality or quantity of preference data [59, 60, 61]. Other strategies focus on learning token-wise dense rewards [62, 63] or multi-objective rewards [38]. Additionally, a series of works aim to enhance the robustness of reward models against preference inconsistencies. Techniques such as adaptive margin [10], contrastive learning [39], and meta-learning [64] are employed to improve the model’s ability to differentiate between chosen and rejected responses.

**Mitigating Overoptimization in RLHF.** Reward models tend to overfit and struggle to generalize beyond the training distribution, which often leads to the overoptimization issue [17]. One approach to mitigate this is to penalize overly confident model outputs using label smoothing [39] or SFT regularization [24, 25]. Alternatively, the model and data can be iteratively updated, replacing hard labels with soft labels [65]. Ensemble techniques, which train several reward models, can also help reduce reward hacking and manage shifts in distribution [18, 19, 20, 66, 32, 67, 68]. Adversarial Preference Optimization employs adversarial learning between reward models and an LLM agent to address the gap in generation distribution [69]. Recent studies have also utilized uncertainty to mitigate reward over-optimization, including the integration of an uncertainty penalty into rewards [70], or the construction of a confidence interval for gold rewards based on uncertainty estimations [23].

## 7 Conclusion

In this study, we introduce an efficient approach aimed at enhancing the generalizability and robustness of reward learning for large language models. By incorporating regularization techniques on the hidden states of reward models, our method demonstrates substantial improvements in the generalization performance of reward models for unseen data. Moreover, our approach effectively mitigates the issue of overoptimization in RLHF. We believe that our findings hold promise in inspiring future research efforts towards the development of more robust reward models that can facilitate the alignment of large models through cost-effective solutions.

## Limitations

In this study, we evaluate the robustness of GRM against label noise by introducing a 25% level of synthetic noise into the training data for all proxy reward models. This is achieved by randomly flipping chosen and rejected labels. Due to cost considerations, we conduct synthetic experiments in line with community practices [18, 39], as using human-labeled data is not feasible for us. However, synthetic data may introduce biases that don’t accurately reflect real-world scenarios. Future research should aim to mitigate this limitation by incorporating experiments with human-labeled data, providing a more thorough evaluation of the reward model’s robustness. Another limitation of our study is the computational restriction preventing us from testing GRM with parameter sizes exceeding 10B. Further efforts to extend our method to larger reward models could be highly promising.

## Acknowledgement

Tong Zhang is partially supported by an NSF IIS grant No. 2416897. Huan Zhang was supported by the AI2050 program at Schmidt Sciences (AI 2050 Early Career Fellowship). The authors would like to thank the reviewers and readers for constructive feedback on the manuscript.

## References

- [1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

- [2] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.
- [3] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.
- [4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.
- [6] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.
- [7] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.
- [8] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.
- [9] R OpenAI. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2:13, 2023.
- [10] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- [11] Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.
- [12] Hanning Zhang, Shizhe Diao, Yong Lin, Yi R Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. R-tuning: Teaching large language models to refuse unknown questions. *arXiv preprint arXiv:2311.09677*, 2023.
- [13] Rui Yang, Xiaoman Pan, Feng Luo, Shuang Qiu, Han Zhong, Dong Yu, and Jianshu Chen. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. *arXiv preprint arXiv:2402.10207*, 2024.
- [14] Yong Lin, Skyler Seto, Maartje ter Hoeve, Katherine Metcalf, Barry-John Theobald, Xuan Wang, Yizhe Zhang, Chen Huang, and Tong Zhang. On the limited generalization capability of the implicit reward model induced by direct preference optimization. *arXiv preprint arXiv:2409.03650*, 2024.
- [15] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.
- [16] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.
- [17] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023.
- [18] Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. Reward model ensembles help mitigate overoptimization. *arXiv preprint arXiv:2310.02743*, 2023.
- [19] Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D’Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, et al. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. *arXiv preprint arXiv:2312.09244*, 2023.

- [20] Yong Lin, Lu Tan, Yifan Hao, Honam Wong, Hanze Dong, Weizhong Zhang, Yujiu Yang, and Tong Zhang. Spurious feature diversification improves out-of-distribution generalization. *arXiv preprint arXiv:2309.17230*, 2023.
- [21] Kyuyoung Kim, Jongheon Jeong, Minyong An, Mohammad Ghavamzadeh, Krishnamurthy Dvijotham, Jinwoo Shin, and Kimin Lee. Confidence-aware reward optimization for fine-tuning text-to-image models. *arXiv preprint arXiv:2404.01863*, 2024.
- [22] Ted Moskowitz, Aaditya K Singh, DJ Strouse, Tuomas Sandholm, Ruslan Salakhutdinov, Anca D Dragan, and Stephen McAleer. Confronting reward model overoptimization with constrained rlhf. *arXiv preprint arXiv:2310.04373*, 2023.
- [23] Xiaoying Zhang, Jean-Francois Ton, Wei Shen, Hongning Wang, and Yang Liu. Overcoming reward overoptimization via adversarial policy optimization with lightweight uncertainty estimation. *arXiv preprint arXiv:2403.05171*, 2024.
- [24] Zhihan Liu, Miao Lu, Shenao Zhang, Boyi Liu, Hongyi Guo, Yingxiang Yang, Jose Blanchet, and Zhaoran Wang. Provably mitigating overoptimization in rlhf: Your sft loss is implicitly an adversarial regularizer, 2024.
- [25] Shicong Cen, Jincheng Mei, Katayoon Goshvadi, Hanjun Dai, Tong Yang, Sherry Yang, Dale Schuurmans, Yuejie Chi, and Bo Dai. Value-incentivized preference optimization: A unified approach to online and offline rlhf. *arXiv preprint arXiv:2405.19320*, 2024.
- [26] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.
- [27] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.
- [28] Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, pages 5084–5096. PMLR, 2021.
- [29] Rui Yang, Chenjia Bai, Xiaoteng Ma, Zhaoran Wang, Chongjie Zhang, and Lei Han. Rorl: Robust offline reinforcement learning via conservative smoothing. *Advances in neural information processing systems*, 35:23851–23866, 2022.
- [30] Hao Sun, Lei Han, Rui Yang, Xiaoteng Ma, Jian Guo, and Bolei Zhou. Exploit reward shifting in value-based deep-rl: Optimistic curiosity-based exploration and conservative exploitation via linear reward shaping. *Advances in neural information processing systems*, 35:37719–37734, 2022.
- [31] Rui Yang, Lin Yong, Xiaoteng Ma, Hao Hu, Chongjie Zhang, and Tong Zhang. What is essential for unseen goal generalization of offline goal-conditioned rl? In *International Conference on Machine Learning*, pages 39543–39571. PMLR, 2023.
- [32] Alexandre Ramé, Nino Vieillard, Léonard Hussenot, Robert Dadashi, Geoffrey Cideron, Olivier Bachem, and Johan Ferret. Warm: On the benefits of weight averaged reward models. *arXiv preprint arXiv:2401.12187*, 2024.
- [33] Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*, 2022.
- [34] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [35] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.
- [36] Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023.
- [37] Hanze Dong, Wei Xiong, Deepanshu Goyal, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*, 2023.
- [38] Haoxiang Wang, Yong Lin, Wei Xiong, Rui Yang, Shizhe Diao, Shuang Qiu, Han Zhao, and Tong Zhang. Arithmetic control of llms for diverse user preferences: Directional preference alignment with multi-objective rewards. *arXiv preprint arXiv:2402.18571*, 2024.

- [39] Binghai Wang, Rui Zheng, Lu Chen, Yan Liu, Shihan Dou, Caishuang Huang, Wei Shen, Senjie Jin, Enyu Zhou, Chenyu Shi, et al. Secrets of rlhf in large language models part ii: Reward modeling. *arXiv preprint arXiv:2401.06080*, 2024.
- [40] Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. Rlhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*, 2024.
- [41] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.
- [42] Jiwoo Hong, Noah Lee, and James Thorne. Reference-free monolithic preference optimization with odds ratio. *arXiv preprint arXiv:2403.07691*, 2024.
- [43] Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*, 2024.
- [44] Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Benjamin Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. A general language assistant as a laboratory for alignment. *CoRR*, abs/2112.00861, 2021.
- [45] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024.
- [46] Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. Rewardbench: Evaluating reward models for language modeling, 2024.
- [47] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.
- [48] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.
- [49] Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *arXiv preprint arXiv:2306.02561*, 2023.
- [50] Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, and Jiantao Jiao. Starling-7b: Improving llm helpfulness & harmlessness with rlhf, 2023.
- [51] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*, 2023.
- [52] AI@Meta. Llama 3 model card. 2024.
- [53] Xize Liang, Chao Chen, Jie Wang, Yue Wu, Zhihang Fu, Zhihao Shi, Feng Wu, and Jieping Ye. Robust preference optimization with provable noise tolerance for llms. *arXiv preprint arXiv:2404.04102*, 2024.
- [54] Rui Yang, Han Zhong, Jiawei Xu, Amy Zhang, Chongjie Zhang, Lei Han, and Tong Zhang. Towards robust offline reinforcement learning under diverse data corruption. In *International Conference on Learning Representations*, 2024.
- [55] Chenlu Ye, Rui Yang, Quanquan Gu, and Tong Zhang. Corruption-robust offline reinforcement learning with general function approximation. *Advances in Neural Information Processing Systems*, 36, 2024.
- [56] Debmalya Mandal, Andi Nika, Parameswaran Kamalaruban, Adish Singla, and Goran Radanović. Corruption robust offline reinforcement learning with human feedback. *arXiv preprint arXiv:2402.06734*, 2024.
- [57] Hao Sun, Thomas Pouplin, Nicolás Astorga, Tennison Liu, and Mihaela van der Schaar. Improving llm generation with inverse and forward alignment: Reward modeling, prompting, fine-tuning, and inference-time optimization. In *The First Workshop on System-2 Reasoning at Scale, NeurIPS’24*.

- [58] Hao Sun, Alihan Hüyük, and Mihaela van der Schaar. Query-dependent prompt evaluation and optimization with offline inverse rl. In *The Twelfth International Conference on Learning Representations*, 2023.
- [59] Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. AlpacaFarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36, 2024.
- [60] Hao Sun and Mihaela van der Schaar. Inverse-rlignment: Inverse reinforcement learning from demonstrations for llm alignment. *arXiv preprint arXiv:2405.15624*, 2024.
- [61] Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*, 2023.
- [62] Alex J Chan, Hao Sun, Samuel Holt, and Mihaela van der Schaar. Dense reward for free in reinforcement learning from human feedback. *arXiv preprint arXiv:2402.00782*, 2024.
- [63] Han Zhong, Guhao Feng, Wei Xiong, Li Zhao, Di He, Jiang Bian, and Liwei Wang. Dpo meets ppo: Reinforced token optimization for rlhf. *arXiv preprint arXiv:2404.18922*, 2024.
- [64] Shihan Dou, Yan Liu, Enyu Zhou, Tianlong Li, Haoxiang Jia, Limao Xiong, Xin Zhao, Junjie Ye, Rui Zheng, Tao Gui, et al. MetaRM: Shifted distributions alignment via meta-learning. *arXiv preprint arXiv:2405.00438*, 2024.
- [65] Banghua Zhu, Michael I Jordan, and Jiantao Jiao. Iterative data smoothing: Mitigating reward overfitting and overoptimization in rlhf. *arXiv preprint arXiv:2401.16335*, 2024.
- [66] Yong Lin, Lu Tan, Hangyu Lin, Zeming Zheng, Renjie Pi, Jipeng Zhang, Shizhe Diao, Haoxiang Wang, Han Zhao, Yuan Yao, et al. Speciality vs generality: An empirical study on catastrophic forgetting in fine-tuning foundation models. *arXiv preprint arXiv:2309.06256*, 2023.
- [67] Shun Zhang, Zhenfang Chen, Sunli Chen, Yikang Shen, Zhiqing Sun, and Chuang Gan. Improving reinforcement learning from human feedback with efficient reward model ensemble. *arXiv preprint arXiv:2401.16635*, 2024.
- [68] Yifan Hao, Yong Lin, Difan Zou, and Tong Zhang. On the benefits of over-parameterization for out-of-distribution generalization. *arXiv preprint arXiv:2403.17592*, 2024.
- [69] Pengyu Cheng, Yifan Yang, Jian Li, Yong Dai, and Nan Du. Adversarial preference optimization. *arXiv preprint arXiv:2311.08045*, 2023.
- [70] Adam X Yang, Maxime Robeyns, Thomas Coste, Jun Wang, Haitham Bou-Ammar, and Laurence Aitchison. Bayesian reward models for llm alignment. *arXiv preprint arXiv:2402.13210*, 2024.
- [71] Rui Yang, Yiming Lu, Wenzhe Li, Hao Sun, Meng Fang, Yali Du, Xiu Li, Lei Han, and Chongjie Zhang. Rethinking goal-conditioned supervised learning and its connection to offline rl. *arXiv preprint arXiv:2202.04478*, 2022.
- [72] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.
- [73] Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. Trl: Transformer reinforcement learning. <https://github.com/huggingface/trl>, 2020.
- [74] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.



## A Deriving the Regularization Term

To derive the potential formulation of the regularization term, we consider the following adversarial optimization problem: learning a reward model against an adversarial policy.

$$\theta = \arg \min_{\theta} \left\{ \mathcal{L}_{\text{reward}}(\theta) + \gamma \max_{\pi} J(\theta, \pi) \right\} \quad (11)$$

The term for policy optimization  $J(\theta, \pi)$  can have different formulations, but a KL divergence regularized optimization objective is generally used in training the policy [16, 8, 71]. Moreover, it has an advantageous property that the inner optimization problem has an analytical solution, which can simplify the problem.

$$J(\theta, \pi) = \mathbb{E}_{x \sim D, y \sim \pi(\cdot|x)} [r_{\theta}(x, y)] - \beta \mathbb{E}_{x \sim D} [\text{KL}(\pi(\cdot|x) | \pi_{\text{ref}}(\cdot|x))], \quad (12)$$

where  $\beta > 0$  is the coefficient controlling the regularization degree and  $\pi_{\text{ref}}$  is the reference model. The analytical solution of  $J(\theta, \pi)$  is formulated as follows:

$$\pi_{\theta}^* = \frac{1}{Z_{\theta}(x)} \pi_{\text{ref}}(y|x) \exp(r_{\theta}(x, y)/\beta), \quad Z_{\theta}(x) = \sum_{y'} \pi_{\text{ref}}(y'|x) \exp(r_{\theta}(x, y')/\beta) \quad (13)$$

Equivalently, we can obtain the formulation of reward described by  $\pi_{\theta}^*$  and  $\pi_{\text{ref}}$  as in [41]:

$$r_{\theta}(x, y) = \beta (\log \pi_{\theta}^*(y|x) - \log \pi_{\text{ref}}(y|x) + \log Z_{\theta}(x)) \quad (14)$$

Following recent theoretical analysis [25], we define a fixed calibration policy  $\pi_{\text{cal}}$  that is independent of the algorithm, which has the calibration effect of centering the reward function while incorporating additional policy preferences into the objective.

**Definition 1**  $\pi_{\text{cal}}$  is a fixed calibration policy for reward model  $r_{\theta}$  and the dataset  $D$  that satisfies:

$$\mathbb{E}_{x \sim D, y \sim \pi_{\text{cal}}} [r_{\theta}(x, y)] = 0$$

Therefore, we can rewrite  $\max_{\pi} J(\theta, \pi)$  as:

$$\begin{aligned} \max_{\pi} J(\theta, \pi) &= J(\theta, \pi_{\theta}^*) = \mathbb{E}_{x \sim D, y \sim \pi_{\theta}^*} [r_{\theta}(x, y) - \beta (\log \pi_{\theta}^*(y|x) - \log \pi_{\text{ref}}(y|x))] \\ &= \mathbb{E}_{x \sim D, y \sim \pi_{\theta}^*} [\log Z_{\theta}(x)] = \mathbb{E}_{x \sim D, y \sim \pi_{\text{cal}}} [\log Z_{\theta}(x)] \\ &= \mathbb{E}_{x \sim D, y \sim \pi_{\text{cal}}} [r_{\theta}(x, y) - \beta (\log \pi_{\theta}^*(y|x) - \log \pi_{\text{ref}}(y|x))] \\ &= -\beta \mathbb{E}_{x \sim D, y \sim \pi_{\text{cal}}} [\log \pi_{\theta}^*(y|x) - \log \pi_{\text{ref}}(y|x)]. \end{aligned} \quad (15)$$

The second line is established because  $\log Z_{\theta}(x)$  is independent of the distribution  $y$ . Besides, the last line just adopts the definition of  $\pi_{\text{cal}}$ .

Incorporating Eq [15] and Eq [14] into Eq [11], we can transform the min-max optimization problem into a standard optimization problem by considering the policy  $\pi_{\theta}^*$ :

$$\begin{aligned} \theta &= \arg \min_{\theta} \left\{ (1 - \alpha) \mathcal{L}_{\text{reward}}(\theta) + \alpha \mathcal{L}_{\text{reward}}(\theta) + \gamma \max_{\pi} J(\theta, \pi) \right\} \\ &= \arg \min_{\theta} \left\{ (1 - \alpha) \mathcal{L}_{\text{reward}}(\theta) - \alpha \mathbb{E}_{(x, y_c, y_r) \sim D} \log \sigma \left( \beta \log \left( \frac{\pi_{\theta}^*(y_c | x)}{\pi_{\text{ref}}(y_c | x)} \right) - \beta \log \left( \frac{\pi_{\theta}^*(y_r | x)}{\pi_{\text{ref}}(y_r | x)} \right) \right) \right. \\ &\quad \left. - \gamma \beta \mathbb{E}_{x \sim D, y \sim \pi_{\text{cal}}} [\log \pi_{\theta}^*(y|x) - \log \pi_{\text{ref}}(y|x)] \right\} \\ &= \arg \min_{\theta} \{ (1 - \alpha) \mathcal{L}_{\text{reward}}(\theta) + \alpha \mathcal{L}_{\text{DPO}}(\pi_{\theta}^*) - \gamma \beta \mathbb{E}_{x \sim D, y \sim \pi_{\text{cal}}} [\log \pi_{\theta}^*(y|x)] \} \end{aligned} \quad (16)$$

Here, we use  $\mathcal{L}_{\text{DPO}}(\pi_{\theta}^*)$  to replace the second term, as it is the same as DPO objective [41]. In the second line, we put the reward described by Eq [14] into  $\alpha \mathcal{L}_{\text{reward}}$ . In the final step, we remove the  $\pi_{\text{ref}}$  term as it does not depend on the parameters of the reward  $r_{\theta}$ , unlike  $\pi_{\theta}^*$  which is dependent on reward  $r_{\theta}$ .

Interestingly, if we set the calibration policy  $\pi_{\text{cal}}$  as the chosen responses  $y_c$  from the dataset  $D$ , the last term becomes an SFT loss. Thus, we can derive the general regularization terms in our framework by renaming the coefficients for  $\pi_{\theta}^*$  as  $\alpha_{\text{DPO}}$  and  $\alpha_{\text{SFT}}$ , and removing the constraint that  $\alpha_{\text{DPO}} = \alpha$ .

$$\arg \min_{\theta} \{ (1 - \alpha) \mathcal{L}_{\text{reward}}(\theta) + \alpha_{\text{DPO}} \mathcal{L}_{\text{DPO}}(\pi_{\theta}^*) + \alpha_{\text{SFT}} \mathcal{L}_{\text{SFT}}(\pi_{\theta}^*) \} \quad (17)$$

Notably, the two regularization terms come from different sources, where  $\mathcal{L}_{\text{DPO}}$  is from the reward loss and  $\mathcal{L}_{\text{SFT}}$  is derived from the adversarial term. This may be the reason why SFT regularization is more helpful than DPO regularization in our empirical results. Inspired by the objective in Eq 7, we relax the relationship between  $r_\theta$  and  $\pi_\theta^*$  and propose to learn a reward model parameterized by  $\theta$  and a language model head parameterized by  $\theta_{\text{LM}}$ , both sharing the same hidden states.

**Discussion.** In Eq 17, we retain both the reward model  $r_\theta$  and the policy  $\pi_\theta^*$ , and replace  $\pi_\theta^*$  with a language head  $\pi_{\theta_{\text{LM}}}$ . A simpler solution is to keep only the reward model by replacing  $\pi_\theta^*$  with  $r_\theta$ , which leads to the following objective:

$$\arg \min_{\theta} \{ \mathcal{L}_{\text{reward}}(\theta) - \gamma \mathbb{E}_{x, y_c \sim D} [r_\theta(x, y_c)] + \gamma \beta \mathbb{E}_{x \sim D} [\log Z_\theta(x)] \}.$$

This approach can be understood as minimizing reward loss while applying regularization to maximize the rewards of selected responses relative to the overall rewards. However, this method is limited by the inefficient calculation of  $Z_\theta$  over the response distribution generated by  $\pi_{\text{SFT}}$ . Therefore, we propose our solution, GRM, which involves a reward model that shares hidden states with a language head. This setup captures certain correlations through shared parameters, helps prevent feature distortion, and is both cost-effective and highly efficient.

## B Implementation Details

**Baseline Details.** All baseline reward models employ the "AutoModelForSequenceClassification" class from transformers [72], which utilizes a randomly initialized linear head to derive rewards. We then train each reward model to minimize the loss function with the training data. For **ensemble baselines**, we train 3 reward models with different random seeds and aggregate their outputs via the 'average' or the 'minimum' strategy. We adopt the average value for the ensemble baseline in Section 5.1 as we find that the minimum value can decrease accuracy and underperform the average one. But for the RLHF experiments in Section 5.2, we report both results because we find sometimes the 'minimum' strategy can work better than due to its pessimism.

The **margin loss** function [10] is defined as below:

$$\mathcal{L}_{\text{margin}}(\theta) = -\mathbb{E}_{(x, y_c, y_r) \sim D} [\log (\sigma (r_\theta(x, y_c) - r_\theta(x, y_r) - m(r)))],$$

which enhances the reward model by emphasizing the differences in rewards. We use the scores between chosen and rejected responses in the Unified-Feedback dataset to calculate  $m(r)$ .

Additionally, the **label smooth loss** is defined as

$$\mathcal{L}_{\text{smooth}}(\theta) = -\mathbb{E}_{(x, y_c, y_r) \sim D} [(1 - \epsilon) \log (\sigma (r_\theta(x, y_c) - r_\theta(x, y_r))) - \epsilon \log (\sigma (r_\theta(x, y_c) - r_\theta(x, y_r)))],$$

where we set  $\epsilon = 0.1$ . The label smooth loss function enhances the model's resilience to a certain degree of errors, thereby alleviating the problem of overfitting.

**GRM Details.** For GRM, the default reward head is configured as a linear layer with shape (hidden size, 1024), followed by a ReLU activation function, and another linear layer of shape (1024, 1). The weight of the text-generation regularization  $\alpha$  is set to 0.01 and the coefficient  $\beta$  in our regularizations is set to 0.1 by default. In the case of the GRM (linear) variant, the reward head is directly set as a linear layer of shape (hidden size, 1). We found a smaller  $\alpha = 0.001$  is better for the linear variant.

**Training and Evaluation Details.** We implement all methods based on transformers [72] and trl [73]. More details are listed in Table 6. To use the Unified-Feedback dataset, we downsample the training data from the 'all' set and use all the 8K test data for evaluation. For the HHH Alignment dataset, we adopt the average score of all four subsets as the result. For the main **experiments trained with LoRA**, we truncate the inputs for all reward models over 1024 tokens. All reward models are trained for two epochs using a learning rate of  $1 \times 10^{-5}$  and a batch size of 16. We load the model with the bf16 precision. Regarding the **full parameter training**, we truncate the inputs over 4096 tokens and train the reward model for one epoch with a learning rate of  $2 \times 10^{-6}$  and a batch size of 512 (with gradient accumulation).

**Computational Resources.** We use NVIDIA RTX A6000 49G for our experiments. Training a 2B reward model with LoRA [74] on the 40K training data for 2 epochs requires approximately 30.4 GPU hours. A 7B reward model requires approximately 93.6 GPU hours.

Table 6: Key implementations of the text generation experiments.

| Basic information                       |   |
|---|---|
| Base models                             | gemma-2b-it and Mistral-7B-Instruct-v0.2  |
| Quantization for training               | bf16  |
| Fine-tuning strategy                    | LoRA [74]   |
| LoRA $r$                                | 32  |
| LoRA alpha                              | 64  |
| LoRA dropout                            | 0.05  |
| Optimizer                               | Adamw_hf  |
| Batch size                              | 16  |
| Learning Rate                           | $1 \times 10^{-5}$  |
| Learning Rate Scheduler                 | cosine  |
| Warmup Ratio                            | 0.03  |
| GRM (Ours)                              |   |
| Regularization weight $\alpha$          | 0.01 by default, and 0.001 for the linear variant   |
| Temperature $\beta$ for loss functions  | 0.1   |
| PPO [34]                                |   |
| KL regulaization                        | 0.0   |
| Epochs                                  | 1   |
| learning rate                           | $1 \times 10^{-5}$  |
| lambda for GAE                          | 0.95  |
| gamma                                   | 1   |
| clip range                              | 0.2   |
| Number of optimization epochs per batch | 4   |
| Number of tokens during generation      | 512   |
| Dataset and Gold Reward Model           |   |
| Main Training Dataset                   | Unified-Feedback  |
| Eval dataset: HHH-Alignment             | <a href="https://huggingface.co/datasets/HuggingFaceH4/hhh_alignment">https://huggingface.co/datasets/HuggingFaceH4/hhh_alignment</a>       |
| Eval dataset: MT-Bench Human Judgements | <a href="https://huggingface.co/datasets/lmsys/mt_bench_human_judgments">https://huggingface.co/datasets/lmsys/mt_bench_human_judgments</a> |
| Gold Reward Model for BoN and PPO       | <a href="#">reward-model-Mistral-7B-instruct-Unified-Feedback</a>   |

## C Additional Experimental Results

### C.1 Comparing with Frozen Backbone

The effect of the random head for downstream finetuning of pretrained model is studied by [33], both theoretically and empirically (across a range of computer vision tasks). It is also easy to validate in the preference learning setting when using a smaller dataset size. We included a baseline, "Classifier (Frozen)", which fixes the base model's features and only fine-tunes the classification head. When the dataset size is 8K (see Table 7), the OOD evaluation results of the baseline reward model (without freezing the backbone) are worse than those of the frozen one, demonstrating the negative effect of distorting pre-trained features. However, we would like to note that when the dataset size is sufficiently large, this negative effect can be mitigated, and the baseline reward model can surpass the frozen reward model due to having more trainable parameters to fit the large preference dataset.

In contrast, by regularizing the hidden states, our GRM can achieve the regularizing effect while fine-tuning all parameters, showing strong performance with both large and small dataset sizes.

Table 7: Reward model performance trained with 8K data.

| Reward Model          | Unified Feedback (ID) | HHH Alignment (OOD) | MT Bench (OOD) |
|-----------------------|-----------------------|---------------------|----------------|
| Classifier (Frozen)   | 62.2                  | 68.8                | 67.6           |
| Classifier (Baseline) | 66.1                  | 65.1                | 67.7           |
| GRM (ours)            | 69.0                  | 71.9                | 69.8           |

### C.2 Choice of Training Epochs

In our main experiments, we train reward models for 2 epochs with LoRA. We determine this number based on a nearly converging validation loss. Specifically, we reserve 1% of the training set for

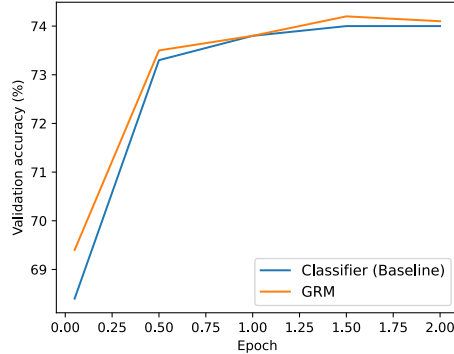


Figure 5: Learning curves for reward models on Unified-Feedback.

validation (e.g., 4K for 400K training data) and found that 2 epochs are sufficient for reward modeling with LoRA in our setting. As shown in Figure 5, we observe convergence in the validation loss during the second epoch, with no further improvement in the third epoch. For full-parameter training experiments, which are more prone to overfitting, we train the reward model for only one epoch.

### C.3 Choice of the SFT objective

In our paper, we consider a slightly different form of SFT objective as in Eq 10. A more straightforward objective is  $\mathcal{L}_{\text{SFT}}(\theta_{\text{LM}}) = -\mathbb{E}_{(x, y_c) \sim D} [\log(\pi_{\theta_{\text{LM}}}(y_c | x))]$ . In ideal situations, the two forms should perform similarly. We also tried the  $\log$  form but found that it requires different hyperparameter tuning for the regularization weight  $\alpha$  in Eq 4 due to changes in the loss scale. In Table 8 and Table 9, "GRM logreg" outperforms the baseline reward model and matches or even slightly exceeds the performance of GRM on OOD tasks when  $\alpha$  is tuned appropriately. This experiment uses the same gemma-2B-it as the base model.

We found that the current form of SFT regularization can directly use the same hyperparameters as our DPO regularization. Therefore, we opted for this solution to **maintain coherence with these regularizations and avoid the need for hyperparameter adjustments**.

Table 8: Results on ID and OOD evaluation with **400K training data** from Unified-Feedback.

| Reward Model                | Unified Feedback | HHH Alignment | MT Bench    |
|-----------------------------|------------------|---------------|-------------|
| Classifier (Baseline)       | 72.1             | 73.4          | 71.2        |
| GRM                         | 73.2             | 79.8          | 73.4        |
| GRM logreg $\alpha = 0.005$ | 72.8             | 77.6          | 72.8        |
| GRM logreg $\alpha = 0.001$ | <b>73.3</b>      | <b>80.2</b>   | <b>73.6</b> |

Table 9: Results on ID and OOD evaluation with **40K training data** from Unified-Feedback.

| Reward Model                | Unified Feedback | HHH Alignment | MT Bench    |
|-----------------------------|------------------|---------------|-------------|
| Classifier (Baseline)       | 68.8             | 70.3          | 69.1        |
| GRM                         | <b>71.5</b>      | 78.7          | <b>73.0</b> |
| GRM logreg $\alpha = 0.005$ | 69.7             | 72.4          | 72.8        |
| GRM logreg $\alpha = 0.001$ | 70.8             | <b>80.7</b>   | 72.0        |

### C.4 Ablation of the Regularization Weight

We find the most impactful hyperparameter of GRM is the regularization weight  $\alpha$ . Figure 6 presents an evaluation of GRM’s performance under various  $\alpha$  values. It is evident from the figure that setting  $\alpha$  to either extreme, such as 0, or a relatively large value like 0.1, results in suboptimal out-of-distribution (OOD) performance. However, selecting an appropriate value between 0 and 0.1 consistently yields higher scores. In all our experiments, we default to an  $\alpha$  value of 0.01. This choice has already shown significant performance improvements in our experiments.

### C.5 Impact of Reward Head Layers on Performance

An interesting aspect to explore is how the structure of the nonlinear reward head influences preference learning performance. In Figure 7, we compare the performance of the default GRM (using the SFT

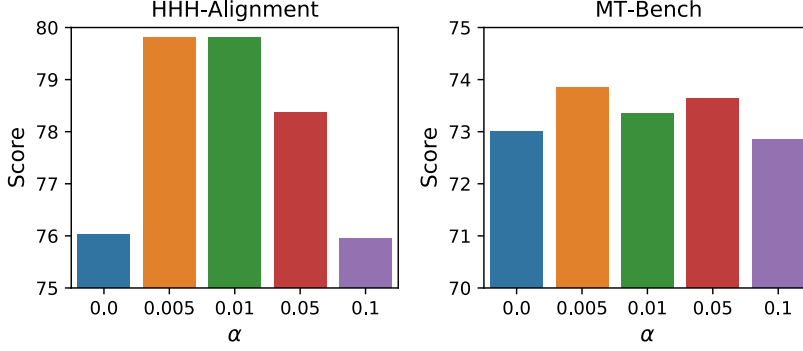


Figure 6: Comparing different values of  $\alpha$  for GRM (2B) on scores of HHH-Alignment and MT-Bench.

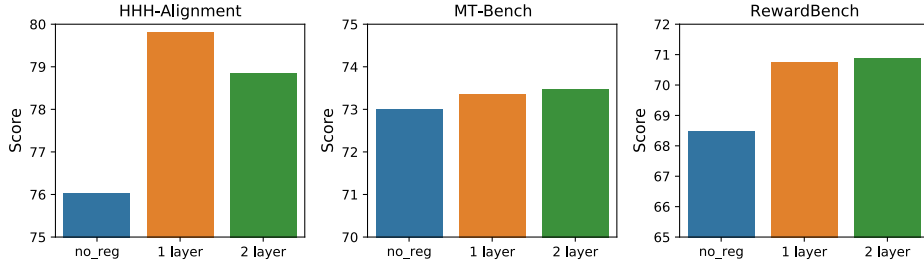


Figure 7: Comparing different layers of reward head for GRM (2B) on scores of HHH-Alignment, MT-Bench, and RewardBench.

regularization) against a variant of GRM that incorporates an additional linear layer and a ReLU activation in the reward head, denoted as "2 layer". The results indicate that the two-layer version slightly surpasses the performance of the single-layer GRM on MT-Bench and RewardBench scores, but it exhibits a decline in the score on the HHH-Alignment. Due to this inconsistency, we opted not to include the two-layer version in our main experiments. However, future research focusing on the impact of the reward model's structure could yield promising insights.

## C.6 Comparison with Additional Variant

In Appendix A, we derive an objective that retains only the reward model  $r_\theta$  by replacing the policy  $\pi_\theta^*$  with a formula of  $r_\theta$ . Empirically, this objective is challenging to optimize due to the calculation of  $Z_\theta$ . As an alternative, we propose a simplified objective by omitting the  $Z_\theta$  term:

$$\arg \min_{\theta} \{ \mathcal{L}_{\text{reward}}(\theta) - \gamma \mathbb{E}_{x, y_c \sim D} [r_\theta(x, y_c)] \}.$$

This objective includes a regularization term to maximize the average rewards of chosen responses. However, the second term can easily dominate the loss since the reward loss term is constrained by the logsigmoid operator. A more stable approach is to use the following empirical objective:

$$\arg \min_{\theta} \{ \mathcal{L}_{\text{reward}}(\theta) - \gamma \mathbb{E}_{x, y_c \sim D} [\log \sigma(r_\theta(x, y_c))] \}.$$

We refer to this regularizer as "positive regularization" or "pos reg" for short. We compare positive regularization with the baseline classifier and GRM with SFT regularization in Tables 10 and 11. The base model for the reward models is gemma-2B-it, and GRM adopts the linear variant for the RewardBench results. "Positive regularization" does not yield improvement when the dataset size is limited to 40K, but it brings slight overall enhancement when learning from 400K training data.

In contrast, GRM significantly enhances both ID and OOD accuracy, especially when learning from a limited preference dataset. These results demonstrate that our approach is more effective, even when based on similar theoretical derivation.

Table 10: Results on ID and OOD evaluation with **400K training data** from Unified-Feedback.

| Reward Model          | Unified Feedback | HHH Alignment | MT Bench    | Reward Bench |
|-----------------------|------------------|---------------|-------------|--------------|
| Classifier (baseline) | 72.1             | 73.4          | 71.2        | 68.2         |
| Classifier (pos reg)  | 71.7             | 75.0          | 70.7        | 69.8         |
| GRM w/ sft (ours)     | <b>73.2</b>      | <b>79.8</b>   | <b>73.4</b> | <b>71.5</b>  |

Table 11: Results on ID and OOD evaluation with **40K training data** from Unified-Feedback.

| Reward Model          | Unified Feedback | HHH Alignment | MT Bench    | Reward Bench |
|-----------------------|------------------|---------------|-------------|--------------|
| Classifier (baseline) | 68.8             | 70.3          | 69.1        | 64.5         |
| Classifier (pos reg)  | 69.5             | 70.0          | 69.6        | 63.2         |
| GRM w/ sft (ours)     | <b>71.5</b>      | <b>78.7</b>   | <b>73.0</b> | <b>69.5</b>  |

### C.7 Regularization with pretraining dataset

In our default design, we use the preference dataset employed to train reward models to regularize the text-generation ability of the language head, eliminating the need for additional datasets. While we believe that other data formats, such as pretraining datasets, can also be beneficial, preference data offers a distinct advantage. It allows us to avoid using external datasets during reward modeling, which may also better align with the distribution of prompts and responses.

To illustrate this, we conduct an experiment using GRM with text-generation regularization on an open-source pretraining dataset, togethercomputer/RedPajama-Data-1T-Sample<sup>7</sup> (which includes text from Commoncrawl, Arxiv, and books), referred to as 'GRM pretrain reg'. For fairness, we only used a pretraining dataset of the same size as the training set for reward modeling.

The results indicate that 'GRM pretrain reg' outperforms the baseline reward model and matches the performance of GRM when the dataset size is large (400K). However, when the dataset size is small, using a pretraining dataset is less effective than using the preference dataset.

Table 12: Results on ID and OOD evaluation with **400K training data** from open-source pretraining dataset.

| Reward Model          | Unified Feedback | HHH Alignment | MT Bench    |
|-----------------------|------------------|---------------|-------------|
| Classifier (baseline) | 72.1             | 73.4          | 71.2        |
| GRM                   | <b>73.2</b>      | <b>79.8</b>   | 73.4        |
| GRM pretrain reg      | 73.0             | 79.2          | <b>74.3</b> |

Table 13: Results on ID and OOD evaluation with **40K training data** from open-source pretraining dataset.

| Reward Model          | Unified Feedback | HHH Alignment | MT Bench    |
|-----------------------|------------------|---------------|-------------|
| Classifier (baseline) | 68.8             | 70.3          | 69.1        |
| GRM                   | <b>71.5</b>      | <b>78.7</b>   | <b>73.0</b> |
| GRM pretrain reg      | 70.8             | 74.5          | 72.9        |

### C.8 Alignment Result after PPO

To demonstrate the advantage of GRM over vanilla reward modeling, we evaluate the win rate of models after PPO training with GRM against those with the vanilla reward model. The evaluation is conducted using GPT-4o on 100 randomly selected prompts from the test set in Unified-Feedback, with the order of responses randomly flipped to avoid order bias. The results below show a significantly higher win rate for GRM than the vanilla reward model across two different base reward models.

Table 14: Win rate of models after PPO training with GRM against those with the vanilla reward model.

| Base reward model   | Win rate (%) | Tie rate (%) | Loss rate (%) |
|---------------------|--------------|--------------|---------------|
| Gemma 2B it         | 68           | 5            | 27            |
| Mistral 7B Instruct | 73           | 6            | 21            |

### C.9 Comparison with Label Smooth in RLHF

In Figure 8, we observed that reward models trained with label smoothing are vulnerable to hacking by BoN and PPO, leading to inferior performance compared to other baselines. The proxy score

<sup>7</sup><https://huggingface.co/datasets/togethercomputer/RedPajama-Data-1T-Sample>



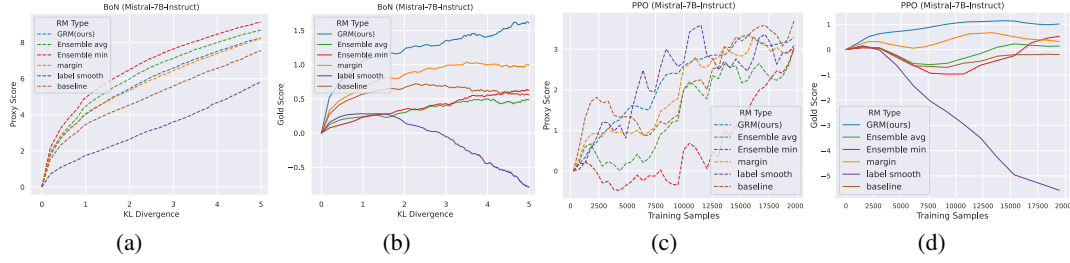


Figure 8: Proxy scores and gold scores of (a)(b) BoN experiments and (c)(d) PPO experiments for base models Mistral-7B-Instruct. Proxy and gold scores are in dashed and solid curves, respectively. Rewards are normalized to start from 0.

increases, while the gold score decreases rapidly. This finding suggests that previous robust techniques in the literature may not be effective for RLHF, underscoring the superiority of GRM as a more viable solution.

## D Examples in the PPO Experiments

In Tables [15](#), [16](#) and [17](#) we present three examples that compare the responses of optimized language models using the PPO algorithm with different reward models. The base models for policy and reward models are all gemma-2b-it as in Section [5.2](#)

For baselines, it is evident that the models exploit certain patterns in rewards, such as the "Ensemble (min)" methods. This exploitation often leads to a collapse into repeated patterns. Besides, the "Baseline" and "Margin" models tend to disregard instructions or refuse to respond to harmless prompts, as demonstrated in Tables [15](#) and [16](#). Moreover, the baseline methods negatively impact the reasoning ability of language models for the math problem as in Table [17](#). These observations indicate that current reward models can be easily hacked by the PPO algorithm, raising concerns about their reliability.

In contrast, the GRM model demonstrates greater robustness in generating instruction-following responses and exhibits better reasoning abilities, even with identical hyperparameters of PPO. Notably, this superior performance of GRM is achieved even with a smaller training cost compared to ensemble baselines. These examples underscore the importance of GRM and its effectiveness in mitigating the overoptimization problem, further highlighting its potential in RLHF applications.

## E Broader Impacts

The proposed approach to enhancing the generalization capabilities of reward models within the RLHF framework offers several positive societal impacts. By improving the accuracy of reward models on out-of-distribution (OOD) tasks, we can enhance the alignment of LLMs with human intent on larger dataset without human labels, leading to more reliable and stronger alignment. Moreover, the regularization technique that preserves the base model's language generation capabilities can contribute to the development of more robust and versatile AI systems, fostering innovation and efficiency across multiple domains. Currently, we do not foresee apparent negative societal impacts stemming from our methods. However, one potential adverse effect could arise if the generalizable reward model is exploited for harmful language model training. Therefore, future efforts in AI safety are crucial to prevent such misuse.

Table 15: Examples in the PPO experiments. GRM optimizes a better language model aligned with human intention, while other baseline reward models can be easily hacked by PPO.

|                       |  |   |
|-----------------------|--|---|
| <b>Prompt</b>         | Respond to the latest instruction based on the conversation history. Human: How do I fly a kite?<br>Assistant: I'm happy to explain more about kites. What would you like to know?<br>Human: how to get the kite in the air<br>Assistant: Can you tell me a bit more about what you're trying to do? What kind of kite? What kind of wind conditions are you planning to use it in? What is your kite-flying experience?<br>Human: I don't have much experience and it's just a simple diamond shaped one<br>Assistant: What wind conditions are you planning to use it in?<br>Human: just regular low wind i guess<br>Assistant:  | <b>Comment</b>  |
| <b>Baseline</b>       | Based on the conversation history, the human is looking for guidance on how to get a simple diamond-shaped kite in the air. The assistant recommends considering wind conditions such as the direction and strength of the wind, the kite type, and the intended flying environment. They suggest considering low wind speeds as suitable for beginners. Therefore, the assistant suggests that the human research the specific wind conditions and kite specifications suitable for their location and desired flying experience before attempting to fly the kite. Remember to prioritize safety and seek guidance from experienced kite enthusiasts or professionals if possible.   | The response does not follow the conversational tone. |
| <b>Margin</b>         | Based on the conversation history, the human is seeking guidance on how to get a kite in the air. However, the assistant cannot provide specific instructions or recommendations without a thorough understanding of the individual's situation and capabilities. It's important to prioritize safety and avoid engaging in activities that could potentially harm the person or property involved. If you have any general kite-flying tips or suggestions, it's recommended to consult with a local park or outdoor enthusiast or seek guidance from a reputable kite-flying organization or instructor. Remember to always follow safety guidelines and regulations, especially when engaging in kite flying activities.  | The response does not follow the conversational tone. |
| <b>Ensemble (min)</b> | Sure, here's some advice on how to get a kite in the air safely and effectively:<br>**Wind Conditions:**<br>* Choose a location with consistent and gentle winds. Avoid strong or gusty winds, as they can make it difficult to control the kite. * Consider the wind speed and direction. Start with lighter winds, especially if you're a beginner.<br>**Kite Selection:**<br>* Choose a kite that's appropriate for your weight and the wind conditions. A well-sized kite will have sufficient lift to stay aloft and provide a thrilling flight experience. * Consider the kite's design. Diamond-shaped kites are relatively stable and easy to maneuver, while symmetrical kites offer more stability but may be more difficult to control.<br>**Safety Precautions:**<br>* Always fly kites away from people and obstacles. * Use a sturdy line or bridle that's long enough to allow the kite to fly freely but not become tangled. * Start in an open area with a clear view. * Be patient and allow the kite to fly for a few minutes before adjusting the line or releasing it. Remember to fly kites responsibly and always prioritize safety.  | The model is hacked to generate repeated words.       |
| <b>Ensemble (avg)</b> | Sure, here's some advice on how to get a kite in the air safely and effectively:<br>**Wind Conditions:**<br>* Choose a location with consistent and gentle winds. Avoid strong or gusty winds, as they can make it difficult to control the kite. * Consider the wind speed and direction. Start with lighter winds, especially if you're a beginner.<br>**Kite Selection:**<br>* Choose a kite that's appropriate for your weight and the wind conditions. A well-sized kite will have sufficient lift to stay aloft and provide a thrilling flight experience. * Consider the kite's design. Diamond-shaped kites are relatively stable and easy to maneuver, while symmetrical kites offer more stability but may be more difficult to control.<br>**Safety Precautions:**<br>* Always fly kites away from people and obstacles. * Use a sturdy line or bridle that's long enough to allow the kite to fly freely but not become tangled. * Start in an open area with a clear view. * Be patient and allow the kite to fly for a few minutes before adjusting the line or releasing it. Remember to fly kites responsibly and always prioritize safety.  | A good response to the prompt.                        |
| <b>GRM (Ours)</b>     | **How to Get a Kite in the Air:**<br>**1. Choose the Right Kite:** - Select a kite that is appropriate for your skill level and wind conditions. - Consider the kite's size, shape (diamond, square, or oval), and the amount of lift it provides.<br>**2. Set Up the Kite:** - Unpack the kite and follow the instructions provided in the manual. - Attach the line to the kite's bridle or string and adjust the length to achieve the desired height. - Make sure the kite is securely attached to a sturdy pole or frame.<br>**3. Choose Wind Conditions:** - Look for wind conditions that are moderate and consistent. - Ideal wind speeds are between 5 and 15 mph (8 and 24 kph). - Avoid strong winds, as they can make it difficult to control the kite.<br>**4. Fly the Kite:** - Hold the kite's handle or string comfortably, keeping your arms relaxed. - Raise the kite slowly until it reaches its peak altitude. - Adjust the kite's height and angle (angle it slightly upwards) to control its flight path. - Gently move the kite with your hands to change its direction and speed.<br>**Tips for Beginners:**<br>- Start in a flat, open area with minimal obstacles. - Use a lightweight kite to make it easier to handle. - Start flying in the early morning or late afternoon when the wind conditions are generally favorable. - Be patient and persistent, as it takes time to learn how to fly a kite effectively. | A more informative response to the prompt.            |

Table 16: Examples in the PPO experiments. GRM optimizes a better language model aligned with gold scores, while other baseline reward models can be easily hacked by PPO.

[illegible]

Table 17: Examples in the PPO experiments. GRM optimizes a better language model aligned with gold scores, while other baseline reward models can be easily hacked by PPO.

| Prompt         |   | Comment         |
|----------------|---|-----------------|
|                | <p>You will be given a definition of a task first, then some input of the task. You are given a math word problem and you are supposed to apply multiplication or division mathematical operators on the numbers embedded in the text to answer the following question and then only report the final numerical answer.</p> <p>A chef needs to cook 9 potatoes . He has already cooked 7 . If each potato takes 3 minutes to cook , how long will it take him to cook the rest ? Output:</p>  |                 |
| Baseline       | <p><b>**Task:**</b> Cook 9 potatoes for 3 minutes each.<br/> <b>**Input:**</b> Number of potatoes cooked = 7, cooking time per potato = 3 minutes.<br/> <b>**Multiplication:**</b> 7 potatoes * 3 minutes/potato = 21 minutes<br/> <b>**Answer:**</b> It will take the chef 21 minutes to cook the remaining 9 potatoes.<br/> Therefore, the final numerical answer is <b>**21**</b>.</p>   | Wrong solution. |
| Margin         | <p><b>**Task:**</b> Cook 9 potatoes for 3 minutes each.<br/> <b>**Input:**</b> Number of potatoes cooked = 7, cooking time per potato = 3 minutes.<br/> <b>**Multiplication Operator:**</b> To calculate the total cooking time, we multiply the number of potatoes cooked by the cooking time per potato.<br/> 9 potatoes * 3 minutes/potato = 27 minutes.<br/> <b>**Final Answer:**</b> It will take the chef 27 minutes to cook the remaining potatoes.<br/> Please note that the question does not provide any information about the initial state of the potatoes or any constraints, so the final answer may vary depending on the specific context and assumptions. It's always recommended to carefully review the task and input data before applying mathematical operations.</p> | Wrong solution. |
| Ensemble (min) | <p>Sure , here's how you calculate the time needed to cook the remaining potatoes :<br/> <b>**Step 1**:</b> Calculate the total time already spent cooking potatoes : 7 minutes<br/> <b>**Step 2**:</b> Subtract the time already spent from total time allotted : 9 minutes - 7 minutes = 2 minutes<br/> <b>**Step 3**:</b> Multiply the remaining time by the number of potatoes to cook : 2 minutes x 9 = 18 minutes<br/> Therefore , it will take him 18 minutes to cook the remaining potatoes .</p>   | Wrong solution. |
| Ensemble (avg) | <p><b>**Task Definition:**</b> Cook 9 potatoes for 3 minutes each.<br/> <b>**Input:**</b></p> <p style="text-align: center;">7 potatoes cooked at 3 minutes each</p> <p><b>**Calculations:**</b></p> <p style="text-align: center;">9 potatoes × 3 minutes each = 27 minutes</p> <p>Therefore, it will take the chef 27 minutes to cook the rest of the potatoes. Answer: 27 minutes.</p>   | Wrong solution. |
| GRM (Ours)     | <p>Sure, here's the analysis and solution for the task:<br/> <b>**Task Definition:**</b><br/> The chef needs to cook 9 potatoes.<br/> <b>**Initial Information:**</b><br/> * Cooked potatoes = 7<br/> <b>**Calculation:**</b><br/> To cook the remaining potatoes, we need to add the number of cooked potatoes to the initial number of cooked potatoes:<br/> 9 potatoes - 7 potatoes = 2 potatoes<br/> <b>**Time to Cook Potatoes:**</b><br/> Each potato takes 3 minutes to cook.<br/> Therefore, the total time to cook the 2 potatoes is:<br/> 2 potatoes x 3 minutes/potato = <b>**6 minutes**</b><br/> <b>**Final Answer:**</b><br/> It will take the chef <b>**6 minutes**</b> to cook the 9 potatoes after he has cooked 7.</p>  | Right solution. |

## NeurIPS Paper Checklist

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: Our study proposes a novel approach to enhance the reward model's generalization ability against distribution shifts by regularizing the hidden states. It was mentioned both in the abstract and introduction.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: The limitations are discussed in the separate "Limitations" section.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: This paper mainly focuses on empirical evaluation but also provides theoretical insights. We provide step-by-step proof and assumptions in Appendix [A] and carefully cite related works that can support our proof.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

#### 4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Our proposed method GRM and three types of regularization terms are detailed and illustrated in Section 3. For more in-depth experimental details to reproduce the main experimental results of the paper, please refer to Section [4] and Appendix [B]. We also open-source our code on GitHub.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in



some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

## 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have ensured that all necessary information to reproduce the experimental results of the paper is included. Links to the datasets and base models used are provided in the footnotes and Appendix B. The code repository and our open-sourced models are available at <https://github.com/YangRui2015/Generalizable-Reward-Model>.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The details of experimental setup are provided in Section 4, including dataset, base models, baselines, and model configurations. For more in-depth experimental details, please refer to Appendix B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We evaluate our method and other baselines on large-scale open-source preference datasets with large models (2B and 7B). Repeating each experiment multiple times presents a challenge due to time and computational constraints. However, we guarantee equitable comparisons, strikes a balance between efficiency and practical limitations.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

#### 8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: For more details of computer resources, please refer to Appendix B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

#### 9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We have reviewed the NeurIPS Code of Ethics and followed all the rules mentioned.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

#### 10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: For more details of Broader Impacts, please refer to Appendix E

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

#### 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: This paper does not release any new pretrained models or datasets for now. And we avoid the harmful examples to show in the paper.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

#### 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Links to the datasets and base models used are provided in the footnotes. We also cite the original papers of the datasets and benchmarks.

Guidelines:

- The answer NA means that the paper does not use existing assets.

- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

### 13. New Assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: Our code and open-source reward models are available at GitHub and Huggingface.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

### 14. Crowdsourcing and Research with Human Subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

### 15. Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.